

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成25年 5月31日現在

機関番号：12612

研究種目：若手研究(B)

研究期間：2010～2012

課題番号：22700026

研究課題名（和文） 機器組込み言語VMのメモリ管理

研究課題名（英文） Memory Management Techniques for Embedded System Language Runtimes

研究代表者

鵜川 始陽 (UGAWA TOMOHARU)

電気通信大学・大学院情報理工学研究科・助教

研究者番号：50423017

研究成果の概要（和文）：本研究では、機器組込み用の言語 VM に適したごみ集めについて研究した。その結果、以下の成果を得た。(1)ごみ集め毎にヒープの一部だけをコンパクトすることで、断片化を一定以下に抑える部分コンパクトを開発し、メモリ不足にならない条件を示した。(2)マルチコア CPU を搭載した組込みシステムで、アプリケーションソフトウェアとは別の CPU コアを使ってごみ集めをする並行ごみ集めを改良した。(3)部分コンパクトを Android に実装する際に問題となったライブラリのバグを修正するためのツールを開発した。(4)ごみ集めのための二つの要素技術を開発した。

研究成果の概要（英文）：In this research, we studied garbage collection techniques for language runtimes on embedded systems. As a result, we achieved the followings. (1) We developed a partial compaction algorithm that ensures that the heap never gets fragmented more than a degree by compacting only a part of the heap in each garbage collection cycle. (2) We improved a concurrent garbage collection so that garbage collector can run efficiently on embedded systems with multi-core CPUs. (3) We developed a program analysis tool that finds bugs in the Android library that were obstacles of our implementation of the partial compaction. (4) We developed two techniques to improve performance of garbage collections.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2010年度	1,100,000	330,000	1,430,000
2011年度	900,000	270,000	1,170,000
2012年度	1,000,000	300,000	1,300,000
年度			
年度			
総計	3,000,000	900,000	3,900,000

研究分野：総合領域

科研費の分科・細目：情報学、ソフトウェア

キーワード：プログラム処理系

1. 研究開始当初の背景

Java は生産性が高い言語として知られており、産業の様々な場面で用いられている。しかし、機器組込みのソフトウェア、特に実

時間処理を含むようなシステムのソフトウェアへの応用は研究段階である。

実時間処理を含む組込みシステムでは、プログラムの長時間にわたる停止を避けるため、ごみ集めを小さな処理に分割して、プロ

グラムと並行して少しずつ進める、実時間ごみ集めを行う必要がある。さらに、限られたメモリで動作し続けるには、空きメモリの断片化に対する対策が必要不可欠になる。しかし断片化を解消するコンパクションの処理はオブジェクトを移動させるため、処理の分割が難しい。

この問題を解決して、実時間処理を含む組込みシステムで Java を利用できるようにする研究が国外では盛んに行われていた。特に EU では JEOPARD プロジェクトという並行実時間システムのための Java 処理系を研究開発するプロジェクトが進められていた。

2. 研究の目的

本研究では、実時間処理を含む組込みシステムを主な対象として、機器組込み言語 VM に必要なごみ集めの研究を行う。特に、実行速度だけでなく、実時間性、信頼性、高いメモリ効率などの特徴を備えたごみ集めを実現するための手法を研究する。具体的には、研究代表者が研究している部分コンパクションのアルゴリズムの改良とそのアルゴリズムの性質を解明、および実装して実証実験を行う。

3. 研究の方法

研究代表者は、プログラムを実行しながら空きメモリの断片化を部分的に解消する、部分コンパクションという技法を研究している。これは、ヒープの連続した一部分を選択して、その部分のオブジェクトを選択領域の外側に移動させることで、連続した空き領域を確保する技法である。同時には一部のオブジェクトしか移動させないので、コンパクション処理の途中も移動元のオブジェクトをそのまま残すことができ、それを利用してプログラムが動き続けることができる。この技法では、(1)いつ、(2)どの大きさの領域を(3)どこに選択するかを決定する必要がある。本研究では、まず断片化が進んでいる領域を効率よく探す方法を開発する。次にコンパクションのタイミングと選択領域の大きさを決める。これらはヒープ全体の大きさやプログラムがどの程度メモリを割り付けるかに依存する。そこで、プログラムのメモリ割り付けに関する振舞いをモデル化し、システムがメモリ不足に陥らないことを保証できるような条件を解明する。さらに、開発したコンパクションを実際の機器組込みシステムに実装して、実験的に性能を測定する。

マルチコア CPU 対応としては、ごみ集め中にミューテータスレッドが動作することを許す、並行ごみ集めを研究する。最終的には部分コンパクションの前段階である実時間

マークスイープごみ集めと部分コンパクションの両方をミューテータスレッドの実行と並行して行うことを目指す。

4. 研究成果

本研究で得られた成果を以下にまとめる。

- (1) 研究代表者が組込み実時間システム向けに開発中の部分コンパクションアルゴリズムを改良し、その性質を調べた。組込みシステムでは利用可能なメモリの量に制限があるため、コンパクションによって空きメモリの断片化を解消し、メモリの利用効率を高めることが必須である。しかしコンパクションの途中にはポインタが正しいアドレスを指さない状態になる期間があり、実用的な性能でミューテータを実行することが難しい。そこで、研究代表者はヒープの一部だけを対象にした部分コンパクションを開発している。

本研究では、部分コンパクションを効率よく行うために、ヒープ中の断片化が進行している領域を選んでコンパクションを行う方法を開発した。さらに、プログラムが使うメモリの量とメモリを割り付ける速さを基に、最低限必要なヒープの大きさと、その大きさのヒープを使った場合に、空きメモリが足りなくならない事を保証できるコンパクションの対象とする領域の大きさや、ごみ集めを開始するタイミングを決定する方法を提案した。

これらの成果はメモリ管理技術に関する国際会議 ISMM2010 および LISP 言語やそれに関連する話題の国際会議 International LISP Conference 2010 において論文が採択され、発表を行った。さらに、提案した手法を組込みシステム用 Java VM である Open VM およびスマートフォン用のプラットフォームである Android で使われている言語 VM に実装して効果を確認した。これらの成果は、それぞれ第 14 回プログラミングおよびプログラミング言語ワークショップと日本ソフトウェア科学会第 28 回大会で発表した。

- (2) 近年のスマートフォンはマルチコア CPU を搭載するようになってきた。そこで、マルチコア CPU を採用した組込みシステムのための並行ごみ集めを検討した。マルチコア CPU では、compare and swap 命令などのアトミックな命令を使うか、メモリバリアを使わなければ、ミューテータとコレクタのスレッド間の通信の完

了が保証できない。並行 GC では書込みバリアにおいてミューテータが書き込んだアドレスをコレクタに伝えることが必要だが、compare and swap 命令やメモリバリア命令には時間がかかり、書込みバリアのように頻繁に実行される処理の中で用いると大幅に性能が低下する。

本研究では、ミューテータスレッド毎に「書込みログ」というバッファを用意し、それを使って書込んだアドレスをまとめてコレクタに渡す方法を提案した。ミューテータは、書込みバリアにおいて書込んだアドレスを書込みログに保存しておく。そして、後でミューテータがコレクタにバッファごとまとめて渡すことで、compare and swap 命令やメモリバリア命令を大幅に削減できる。

この方法では、コレクタがミューテータから渡された書込みログを処理している間に、ミューテータは新しい書込みログにアドレスを書き込んでいるため、書込みログを渡す処理は何回か行う必要がある。本研究では、この処理を終了するための条件を示し、Android の言語 VM に実装して、実験により現実的な回数の繰り返しで終わることを確かめた。

この成果をまとめた論文がコンピュータソフトウェア誌に採録された。

- (3) 我々が研究の対象とした Android は Java のクラスファイルから変換されたバイトコードを実行する言語 VM である Dalvik VM を搭載している。Dalvik VM では通常の Java VM と同様に Java Native Interface (JNI) を介して C 言語や C++ 言語で書かれたモジュールを読み込み、Java のプログラムから呼び出すことができる。Android では、この機能を利用して Android アプリが用いるフレームワークライブラリを提供している。しかし、フレームワークライブラリには JNI を誤って使っている箇所が何箇所もあった。そのため、Dalvik VM にコンパクションを実装して実験することができなかった。そこで本研究では C または C++ 言語で書いたプログラム中の、JNI を誤って使っている可能性のある箇所を指摘するツールを作成した。

この成果をまとめた論文は情報処理学会論文誌：プログラミングに採録された。さらに、本研究で作成したツールを用いて Android のフレームワークライブラリ中のバグを発見して修正した。

この成果は第 15 回プログラミングおよびプログラミング言語ワークショップで発表した。

- (4) 組み込みシステムでは、機器の設計時にその上で動作するソフトウェアが必要とするリソースの上限を見積る必要がある。しかし、ごみ集めのアルゴリズムとして一般的なマークスイープごみ集めは、ごみ集めルーチンがヒープの中を探索するために使うスタックの大きさを現実的な大きさで抑えることができなかった。そのため、まれに非常に多くのメモリを必要とする可能性があった。スタックを用いずに探索する方法も提案されているが、スタックを用いる方法に比べて非常に遅い問題があった。

本研究では、非常に小さいが、ほとんどの場合をカバーできるのに十分な大きさのスタックだけで、ほとんど実行速度を落とさずに探索する方法を提案した。そのスタックが溢れた時には、スタックに詰めなかったオブジェクトの凡その位置を別の表に記録しておき、それを手掛かりに後から再度探索する。表は並行スイープごみ集めで一般的に使われているものを利用するため、追加でメモリを割り当てる必要はない。

この成果は論文が情報処理学会論文誌：プログラミングに採録され、情報処理学会から山下記念研究賞を受賞した。

- (5) 多くのごみ集めアルゴリズムで共通する処理に、ヒープ全体をスキャンして生きている印が付いたオブジェクトを探す処理がある。本研究では、この処理を高速に行う方法を提案した。

既存のヒープスキャンの手法には、ヒープを直接スキャンする方法 (HS) と、オブジェクトが生きている印をヒープの外のビットマップに記録し、それをスキャンする方法 (BS) がある。本研究では、生きているオブジェクトの粗密によって HS と BS のどちらが高速にスキャンできるかが変わることを発見した。そこで、スキャンの途中でオブジェクトの粗密に基づいて HS と BS を切り替える、adaptive scan という手法を開発した。

この成果は情報処理学会プログラミング研究会で発表した。また、メモリ管理技術に関する国際会議 ISMM 2013 において論文が採択された。会議の開催は 2013 年 6 月の予定である。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 4 件)

- ① Haruna Nishiwaki, Tomoharu Ugawa,

Seiji Umatani, Masahiro Yasugi, Taiichi Yuasa: SEAN: Support Tool for Detecting Rule Violations in JNI Coding, IPSJ Transactions on Programming, Vol. 5, No. 3, pp. 23--28, 2012,

<http://id.nii.ac.jp/1001/00083505/>

- ② 鵜川始陽, 信岡孝佳, 海野弘成, 湯淺太一: 書込みバリアにロックやメモリバリア命令を用いない並行スナップショットごみ集め. コンピュータソフトウェア, Vol. 29, No. 3, pp. 143-156, 2012, DOI: 10.11309/jssst.29.3_143
- ③ Tomoharu Ugawa, Hideya Iwasaki, Taiichi Yuasa: Improvements of Recovery from Marking Stack Overflow in Mark Sweep Garbage Collection. IPSJ Transaction on Programming, Vol. 5, No. 1, pp. 1--8, 2012, <http://id.nii.ac.jp/1001/00081552/>
- ④ 鵜川始陽, 組込み Java の実時間ごみ集め, コンピュータソフトウェア, 査読有, Vol. 29, No. 1, pp. 2-18, 2012, DOI: 10.11309/jssst.29.1_2

[学会発表] (計 8 件)

- ① Kazuya Morikawa, Tomoharu Ugawa, Hideya Iwasaki: Adaptive Scanning Reduces Sweep Time for the Lisp2 Mark-Compact Garbage Collector, The 2013 ACM SIGPLAN International Symposium on Memory Management (ISMM 2013), 査読有, 20 June 2013, Seattle USA, 採録済
- ② 西脇春名、鵜川始陽、馬谷誠二、八杉昌宏、湯淺太一、JNI 規則違反検出ツール SEAN を用いた Android のバグ修正、第 15 回プログラミングおよびプログラミング言語ワークショップ(PPL 2013)、査読有、セッション 8、2013 年 3 月 5 日、福島県
- ③ 森川和哉、鵜川始陽、岩崎英哉、ビットマップマーキングを利用したマークコンパクトごみ集めの Jikes RVM への実装、情報処理学会プログラミング研究会、査読無、2012 年 6 月 21 日、北海道
- ④ Haruna Nishiwaki, Tomoharu Ugawa, Seiji Umatani, Masahiro Yasugi, Taiichi Yuasa, Detecting Bugs in Android Using a Static Escape Analyzer SEAN for Native Code, 10th Asian Symposium on Programming Languages and Systems (APLAS 2012), 査読無, poster presentation, 11-13 December 2012, Kyoto Japan
- ⑤ 加藤瞬、鵜川始陽、岩崎英哉、部分コンパクトションの実行頻度と Evacuation

Area 選択法の改良、第 14 回プログラミングおよびプログラミング言語ワークショップ(PPL2012)、査読無、ポスターセッション、2012 年 3 月 8 日、和歌山県

- ⑥ 海野弘成、鵜川始陽、信岡孝佳、湯淺太一、Dalvik VM におけるインクリメンタルコンパクトションの実装と評価、日本ソフトウェア科学会第 28 回大会、査読無、発表番号 5E-1、2011 年 9 月 29 日、沖縄県
- ⑦ Tomoharu Ugawa, Hideya Iwasaki, and Taiichi Yuasa, Starvation-free Heap Size for Replication-Based Incremental Compacting Garbage Collection, The 2010 International LISP Conference (ILC 2010), 査読有, pp. 43-50, 20 October 2010, Reno USA, DOI: 10.1145/1869643.1869649
- ⑧ Tomoharu Ugawa, Hideya Iwasaki, Taiichi Yuasa, Improved Replication-Based Incremental Garbage Collection for Embedded Systems, The 2010 International Symposium on Memory Management (ISMM 2010), 査読有, pp. 73-82, 5 June 2010, Toronto Canada, DOI: 10.1145/1806651.1806664

6. 研究組織

(1) 研究代表者

鵜川 始陽 (UGAWA TOMOHARU)

電気通信大学・大学院情報理工学研究所・助教

研究者番号 : 50423017